REMARKS/ARGUMENTS

Claim 1 has been amended to further clarify that the change in the flow control is between the time a command is likely to place a reference to an object on an execution stack and the time the reference is used to access the object (see, for example, Fig. 8). In addition, claim 1 has been amended to further clarify that the reference stack is not used to execute computer program code and is designated to store only references to objects which have been stored in a heap (see, for example, Fig. 2). Furthermore, claim 9 has been amended to further clarify that a reference is a direct reference to an object which has been stored in a heap designated for storing objects (see, for example, Fig.1). Other independent claims have been amended in a similar manner.

It is very respectfully submitted that determining whether at least two control paths lead to a common subroutine (see, Figs. 1 and 2 of *Agesen et al.*), does NOT teach or even remotely suggest determining whether there is a change in the flow control between the time a command is likely to place a reference to an object on an execution stack and the time the reference is used to access the Java object.

It is noted that *Steele Jr.* states that a stack is implemented "as two substacks, one to contain references and one to contain primitive data." (Steele Jr., Abstract). However, it is respectfully submitted that segregating the contents of a stack into substacks does NOT teach a reference stack which is not used to execute computer program code and is designated to store only references to objects which have been stored in a heap.

The Examiner's rejection is further traversed for these additional reasons.

(a) The verifier described by *Steele Jr.* does NOT teach or suggest: determining whether a Java command is likely to place the only reference to a Java object on an execution stack (Claim 9)

In the Final Office Action, the Examiner has asserted that *Steele Jr.* teaches: "determining the number of references to an object" (Final Office Action, page 2). It is noted that *Steele Jr.* teaches: "number of reference variables (R) is determined" (*Steele Jr.*, Col. 16, lines 48-50). However, contrary to the Examiner's assertion, it is very

respectfully submitted that determining the number of reference variables does NOT teach or suggest: determining the number of references to an object (i.e., the same object).

As depicted in Fig. 1 below, determining the number of references (e.g., R_1, R_2 and R_3) is different than determining whether a reference (e.g., R3) is the only reference to an object (e.g., O_2). It should also be noted that a reference is a direct reference to an object which has been stored in a heap designated for storing objects. Clearly, *Steele Jr.* is referring to variable slots used by a method (*Steele Jr.*, Col. 15, line 3), and NOT the number of references to a particular object. As such, it is respectfully submitted that *Steele Jr.* does NOT teach or suggest determining whether a Java command is likely to place the only reference to a Java object on an execution stack because, among other things, it fails to even teach: determining the number of references to a particular object.

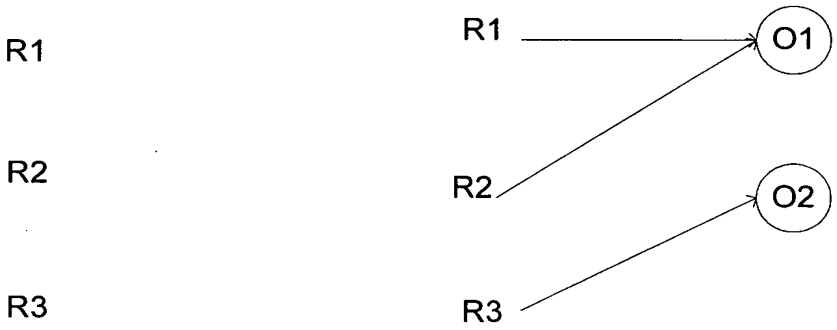

Fig. 1

(b) Determining control paths taught by *Agesen et al.* does NOT teach or suggest determining whether there is a change in the flow control (Claim 1)

It is noted that *Agesen et al.* teaches: "determining whether an instruction sequence includes code defining at least two control paths leading to a common jsr subroutine" (*Agesen et al.*, Col. 12, lines 55-59). However, contrary to the Examiner's assertion, it is very respectfully submitted that determining whether at least two control

paths lead to a common subroutine (see, Figs. 1 and 2 of *Agesen et al.*), does NOT teach or suggest determining whether there is a change in the flow control (Please see Fig. 2 below).

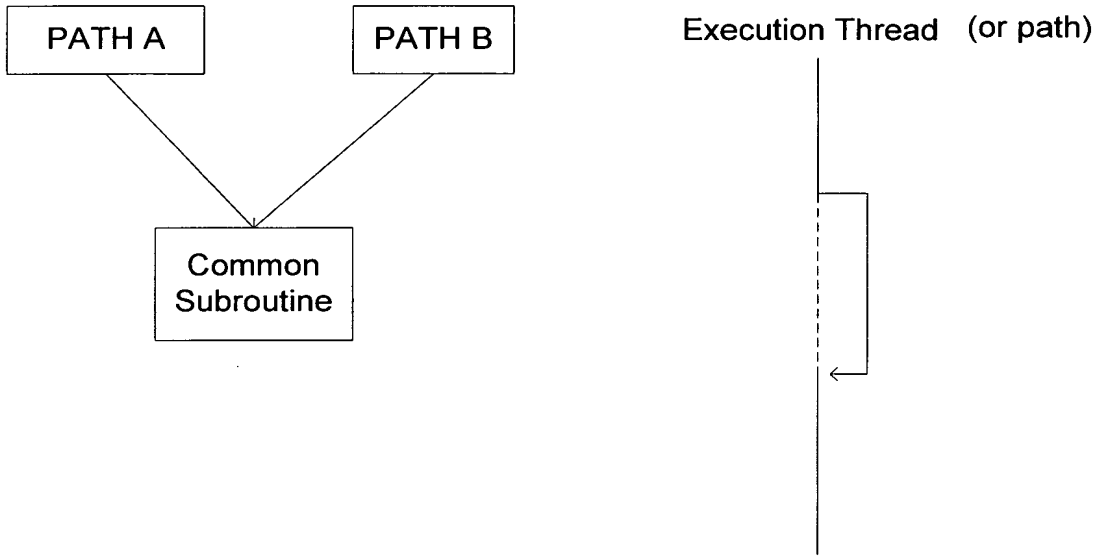

Fig. 2

(c) *Agesen et al.* does NOT teach or suggest determining whether there is a change in flow control in connection with determining whether a command is likely to place a reference to an object on an execution stack (Claim 1)

Agesen et al. et al. pertains to removal of reference conflicts (*Agesen et al.*, Title). It is respectfully submitted that *Agesen et al.* does NOT teach or suggest: determining control paths for the purpose or even in connection with determining: when an instruction is likely to place a reference on an execution stack. Accordingly, it is respectfully submitted that *Agesen et al.* cannot possibly teach or suggest this feature and claim 1 is therefore believed to be patentable over *Agesen et al.* for this additional reason.

Furthermore, in order to combine the references, The Examiner needs to show that *Agesen et al.* at least suggest a connection between the change in flow control in connection with determining whether a command is likely to place a reference to an object on an execution stack. In the Advisory Action, the Examiner has asserted that

Agesen et al. teaches determining whether a variable is a reference. Contrary to the Examiner's assertion, *Agesen et al.* states that "When it is determined that an instruction sequence t includes code defining at least two control paths leading to a common jsr subroutine and it is not possible to determine whether a variable from both paths is a reference or non-reference, and that variable is used after return from the subroutine on at least one of the paths, one or more instructions, including those from one of the paths, are rewritten to eliminate the use of the same variable as the other path and, with it, the conflict itself" (*Agesen et al., Col 12, lines 56-64).* Thus, the teaching of *Agesen et al.* relates to situations where it is NOT possible to determine whether a variable from both paths is a reference or non-reference.

(d) The Examiner has NOT made an prima facie case of obviousness because the Examiner has failed to provide a motivation or suggestion for combining *Agesen et al.* and *Steele Jr.*

In the Advisory Action, the Examiner has asserted that the combination of references would have been obvious because both references are concerned with garbage collection. It is respectfully submitted that the mere fact that both references may have a general goal absent a specific motivation or suggestion in the references for combining them is NOT enough to make a prima facie case of obviousness.

Accordingly, it is very respectfully submitted that general allegation that *Agesen et al* and *Steele Jr.* can be combined to improve garbage collection is NOT enough to establish a prima facie case of obviousness (see, for example, MPEP §2143.01, paragraphs 1 and 3). The Examiner needs to provide a motivation or suggestion in the references themselves, or in the general art, for combining the references in the first place. In this case, the Examiner has failed to provide a motivation or suggestion for combining *Agesen et al.* and *Steele Jr.* as the Examiner has merely made a general allegation that the combination would improve garbage collection.

Moreover, in this case, there is no motivation to combine the reference because, among other things, *Agesen et al.* pertains to "removal of reference conflicts" and not determining whether a command is likely to place a reference to an object on an execution stack. Again, eliminating conflicts associated with control paths is NOT the same goal or motivation for determining whether a command is likely to place a

reference on an execution stack. As such, there is NO need or motivation to combine the analysis of control path in *Agesen et al.* with the teaching of *Steele Jr.*

CONCLUSION

Based on the foregoing, it is submitted that all the pending claims are patentably distinct over the cited art of record. Additional limitations recited in the independent claims or the dependent claims are not further discussed because the limitations discussed above are sufficient to distinguish the claimed invention from the cited art. Accordingly, Applicant believes that all pending claims are allowable and respectfully requests a Notice of Allowance for this application from the Examiner.

Applicants hereby petition for an extension of time which may be required to maintain the pendency of this case, and any required fee for such extension or any further fee required in connection with the filing of this Amendment is to be charged to Deposit Account No. 500388 (Order No. SUN1P833). Should the Examiner believe that a telephone conference would expedite the prosecution of this application, the undersigned can be reached at the telephone number set out below.

Respectfully submitted,
BEYER WEAVER & THOMAS, LLP

R. Mahboubian
Reg. No. 44,890

P.O. Box 70250
Oakland, CA 94612-0250
(650) 961-8300